

第9章 输入输出系统



目录

1

9.1主机与外设的连接

2

9.2程序查询方式及其接口

3

9.3中断系统和程序中中断方式

4

9.4DMA方式及其接口

5

9.5通道控制方式



目录

1

9.1主机与外设的连接

2

9.2程序查询方式及其接口

3

9.3中断系统和程序中中断方式

4

9.4DMA方式及其接口

5

9.5通道控制方式

前言

计算机的输入输出系统是整个计算机系统中最具有多样性和复杂性的部分，本章首先介绍主机与外设之间的连接问题，接着重点介绍程序查询方式、程序中断方式、DMA方式和通道方式。



9.1.1 输入输出接口

- 输入/输出接口（I/O接口）是主机和外设之间的交接界面，通过接口可以实现主机和外设之间的信息交换。
- 主机和外设各自具有自己的工作特点，它们在信息形式和工作速度上具有很大的差异，接口正是为了解决这些差异而设置的。





9.1.1 输入输出接口

数据信息

控制信息

状态信息

联络信息

外设识别信息





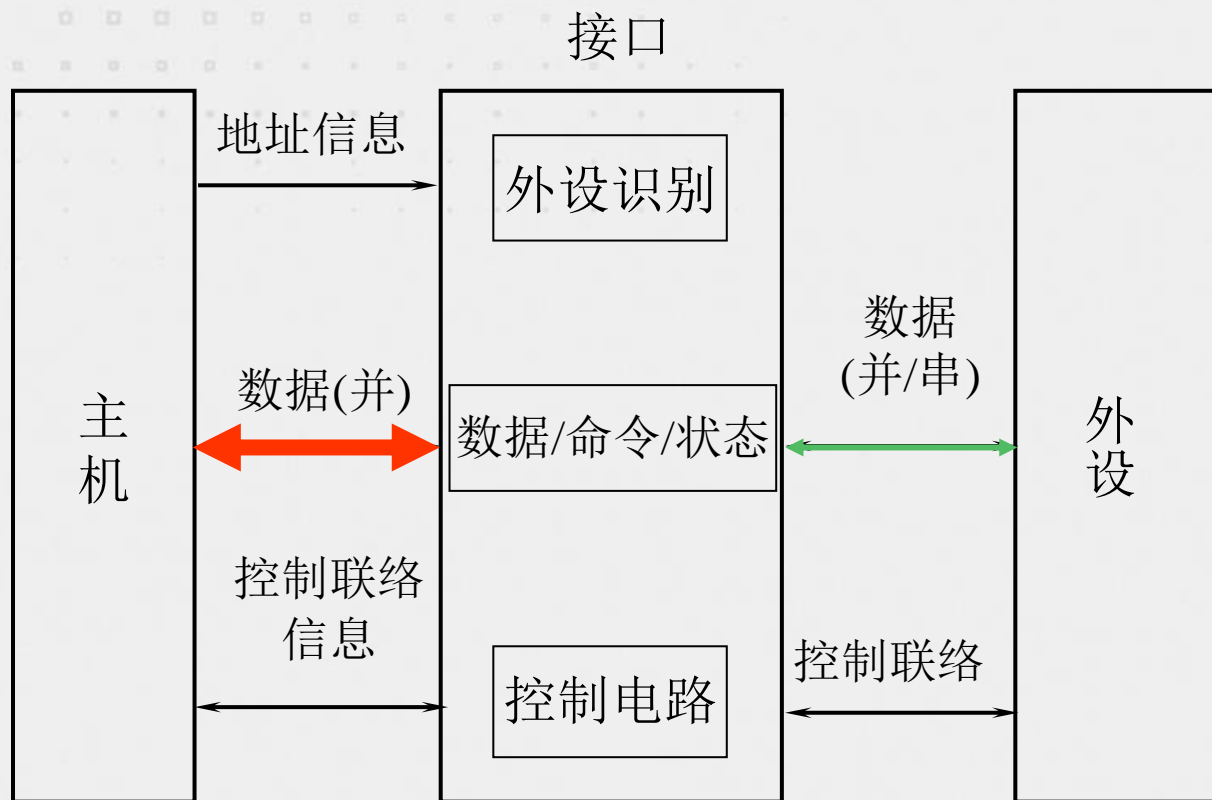
- 1. 接口的功能
 - (1) 实现主机和外设的通信联络控制
 - (2) 进行地址译码和设备选择
 - (3) 实现数据缓冲
 - (4) 数据格式的变换
 - (5) 传递控制命令和状态信息





9.1.2 接口的功能和基本组成

- 接口的基本组成





- 接口与端口是两个不同的概念。端口是指接口电路中可以进行读/写的寄存器，若干个端口加上相应的控制逻辑电路才组成接口。
- 通常，一个接口中包含有数据端口、命令端口和状态端口。





- 3. 接口的类型
 - (1) 按数据传送方式分类
 - (2) 按主机访问I/O设备的控制方式分类
 - (3) 按功能选择的灵活性分类
 - (4) 按通用性分类
 - (5) 按输入/输出的信号分类
 - (6) 按应用来分类





- 1. 端口地址编址方式

I/O映射方式（独立编址）

- 必须设置专门的I/O指令

存储器映射方式（统一编址）

- 不设置专门的I/O指令





- 2.独立编址方式的端口访问
 - Intel 80x86最多可直接寻址256个字节端口，可间接寻址65536 个字节端口。
 - 任意两个连续的8位端口可作为16位端口处理；四个连续的8位端口可作为32位端口处理。





9.1.4

输入/输出信息传送控制方式

程序查询方式

程序中断方式

直接存储器存取（DMA）方式

I/O通道控制方式





- 1. 程序查询方式
 - 程序查询方式是一种程序直接控制方式，这是主机与外设间进行信息交换的最简单方式，输入和输出完全是通过CPU执行程序来完成的。





- 2. 程序中断方式

- 外在作好输入/输出准备时，向主机发中断请求，主机接到请求后就暂时中止原来执行的程序，转去执行中断服务程序对外部请求进行处理，在中断处理完毕后返回原来的程序继续执行。





- 3. 直接存储器存取（DMA）方式
 - DMA方式是在主存储器和外部设备之间开辟直接的数据通路，可以进行基本上不需要CPU介入的主存和外设之间的信息传送，这样不仅能保证CPU的高效率，而且能满足高速外设的需要。





- 4. I/O通道控制方式
 - 通道是一个具有特殊功能的处理器，它能独立地执行通道程序，产生相应的控制信号，实现对外设的统一管理和外设与主存之间的数据传送。





目录

1

9.1主机与外设的连接

2

9.2程序查询方式及其接口

3

9.3中断系统和程序中中断方式

4

9.4DMA方式及其接口

5

9.5通道控制方式



- 程序查询的基本思想
 - 由CPU执行一段输入、输出程序来实现主存与外设之间的数据传送方式，叫做程序直接控制方式。根据外设的不同性质，这种传送方式又可分为无条件传送和程序查询方式两种。





- 最简单、经济的I/O方式，只需很少的硬件。
- 通常接口中至少有两个寄存器

数据端口

状态端口



目录

1

9.1主机与外设的连接

2

9.2程序查询方式及其接口

3

9.3中断系统和程序中中断方式

4

9.4DMA方式及其接口

5

9.5通道控制方式



- 1. 中断的提出

- 程序查询方式存在着下列明显的缺点。

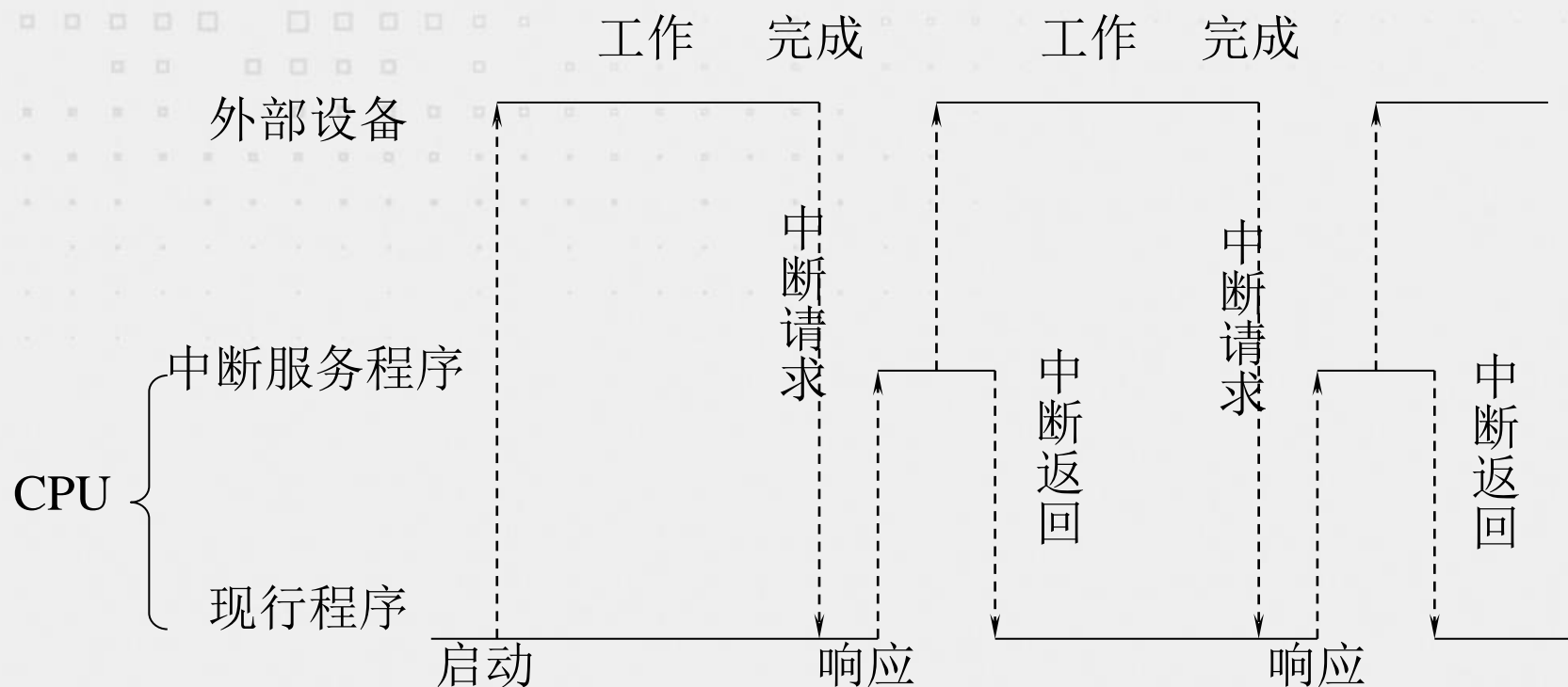
- ✘ ① 在查询过程中，CPU长期处于踏步等待状态，使系统效率大大降低。
 - ✘ ② CPU在一段时间内只能和一台外设交换信息，其它设备不能同时工作。
 - ✘ ③ 不能发现和处理预先无法估计的错误和异常情况。





9.3.1

中断的基本概念





- 2. 程序中断与调用子程序的区别
 - 表面上看起来，计算机的中断处理过程有点类似于调用子程序的过程，这里现行程序相当于主程序，中断服务程序相当于子程序。但是，它们之间却是有着本质上的区别的。





- (1) 子程序的执行是由程序员事先安排好的（由一条调用子程序指令转入），而中断服务程序的执行则是由随机的中断事件引起的；
- (2) 子程序的执行受到主程序或上层子程序的控制，而中断服务程序一般与被中断的现执行程序毫无关系；
- (3) 不存在同时调用多个子程序的情况，而有可能发生多个外设同时请求CPU为自己服务的情况。





- 3. 中断的基本类型
 - (1) 自愿中断和强迫中断
 - (2) 程序中断和简单中断
 - (3) 内中断和外中断
 - (4) 向量中断和非向量中断
 - (5) 单重中断和多重中断





- 1. 中断源和中断请求信号

- 中断源是指中断的来源，即任何引起计算机中断的事件，一般计算机都有多个中断源。由于每个中断源向CPU发出中断请求的时间是随机的，为了记录中断事件并区分不同的中断源，可采用具有存储功能的触发器来记录中断源，称为中断请求触发器。当某一个中断源有中断请求时，其相应的中断请求触发器置成“1”状态，此时，该中断源向CPU发出中断请求信号。





- 1. 中断源和中断请求信号
 - 多个中断请求触发器构成一个中断请求寄存器，其中每一位对应一个中断源，中断请求寄存器的内容称为中断字或中断码，中断字中为“1”的位就表示对应的中断源有中断请求。





- 中断请求信号的传送
 - (1) 独立请求线
 - (2) 公共请求线
 - (3) 二维结构





- 3. 中断优先级与判优方法
 - 当多个中断源同时发出中断请求时，CPU在任何瞬间只能接受一个中断源的请求。通常，把全部中断源按中断的性质和处理的轻重缓急安排优先级，并进行排队。





- 1. CPU响应中断的条件
 - (1) CPU接收到中断请求信号
 - (2) CPU允许中断
 - (3) 一条指令执行完毕





- 2. 中断隐指令

- CPU响应中断之后，经过某些操作，转去执行中断服务程序。 这些操作是由硬件直接实现的，我们把它称为中断隐指令。中断隐指令并不是指令系统中的一条真正的指令，它没有操作码，所以中断隐指令是一种不允许、也不可能为用户使用的特殊指令。其所完成的操作主要有：





- 2. 中断隐指令
 - 中断隐指令所完成的操作主要有：
 - (1) 保存断点
 - (2) 暂不允许中断
 - (3) 引出中断服务程序





- 3. 中断周期

- 中断周期需完成如下操作：

- ⌘ (1) 将特定地址 “0” 送至存储器地址寄存器，记作 $0 \rightarrow \text{MAR}$ ；
 - ⌘ (2) 将PC的内容（断点）送至MDR，记作 $(\text{PC}) \rightarrow \text{MDR}$ ；
 - ⌘ (3) 向主存发写命令，启动存储器做写操作，记作Write；
 - ⌘ (4) 将MDR的内容通过数据总线写入到MAR所指示的主存单元（0号）中，记作 $\text{MDR} \rightarrow \text{M}(\text{MAR})$ ；





- 3. 中断周期

- 中断周期需完成如下操作：

- ⌘ (5) 向量地址形成部件的输出送至PC，为进入中断服务程序作准备，记作向量地址
→PC；

- ⌘ (6) 关中断，将中断允许触发器清0，记作0→EINT。

- 如果断点存入堆栈，只需将上述(1)改为堆栈指针SP→MAR。

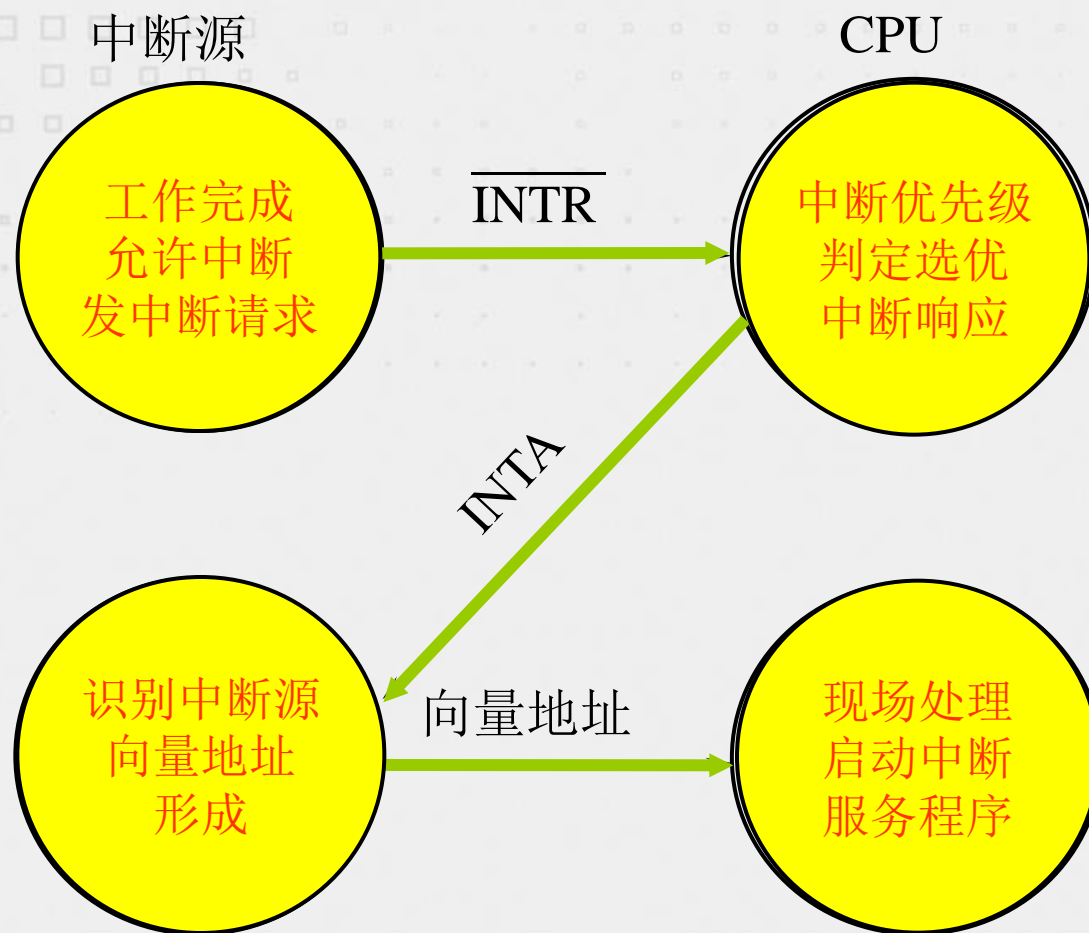




- 4. 进入中断服务程序

- 识别中断源在于转入为该中断源专门设置的中断服务程序。
- 向量中断时，中断源向CPU发出中断请求信号之后，CPU经过一定的判优处理，若决定响应这个中断请求，则向中断源发出中断响应信号。中断源接到中断响应信号后就通过自己的向量地址发生器向CPU发送向量地址。







- 5. 中断现场的保护和恢复

- 中断现场指的是发生中断时CPU的主要状态，其中最重要的是断点，另外还有一些通用寄存器的状态。之所以需要保护和恢复现场的原因是因为CPU要先后执行两个完全不同的程序（现行程序和中断服务程序），必须进行两种程序运行状态的转换。





- 5. 中断现场的保护和恢复

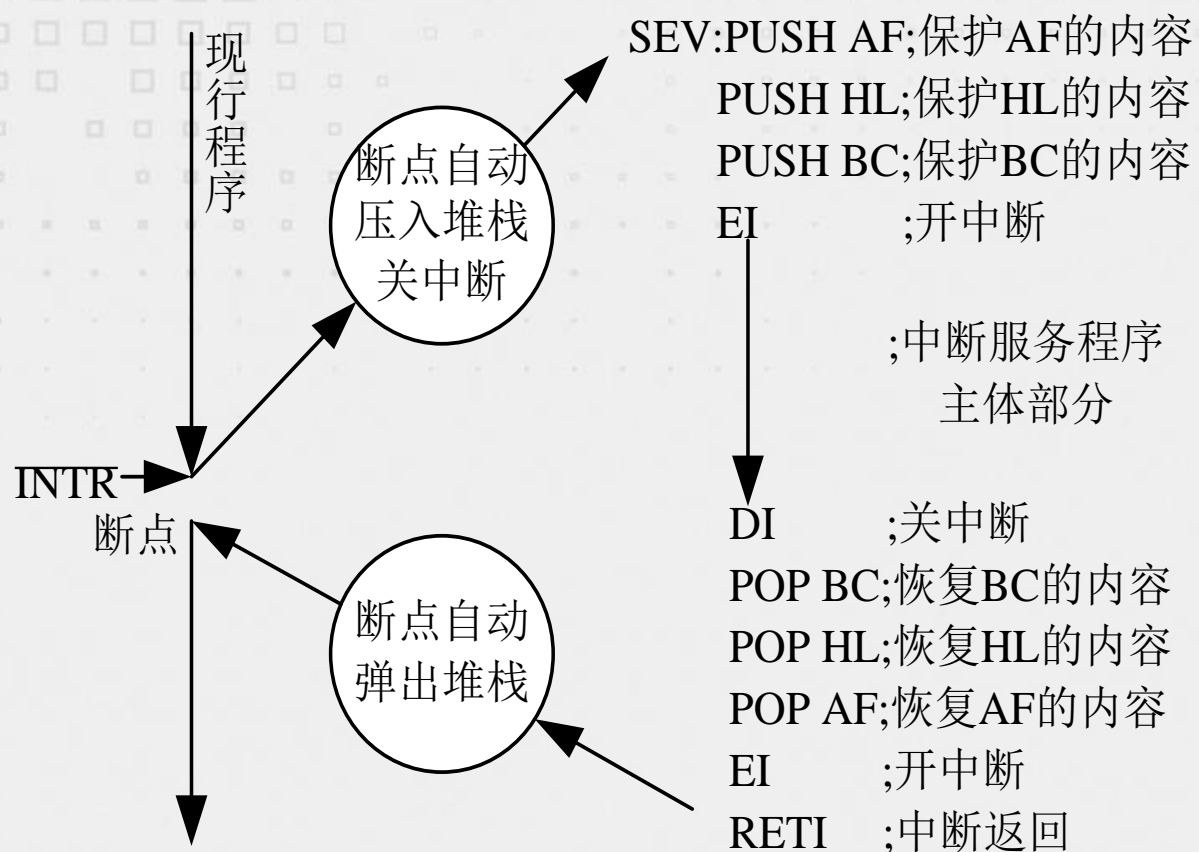
- 一般来说，在中断隐指令中，CPU硬件将自动保存断点，有些计算机还自动保存程序状态寄存器的内容。但是，在许多应用中，要保证中断返回后原来的程序能正确地继续运行，仅保存这一、二个寄存器的内容是不够的。
 -





- 5. 中断现场的保护和恢复
 - 为此，在中断服务程序开始时，应由软件去保存那些硬件没有保存，而在中断服务程序中又可能用到的寄存器（如某些通用寄存器）的内容，在中断返回之前，这些内容还应该被恢复。
 - 现代计算机一般都先采用硬件方法来自动快速的保护和恢复部分重要的现场，其余寄存器的内容再由软件完成保护和恢复，这种方法的硬件支持是堆栈。







- 1. 中断嵌套

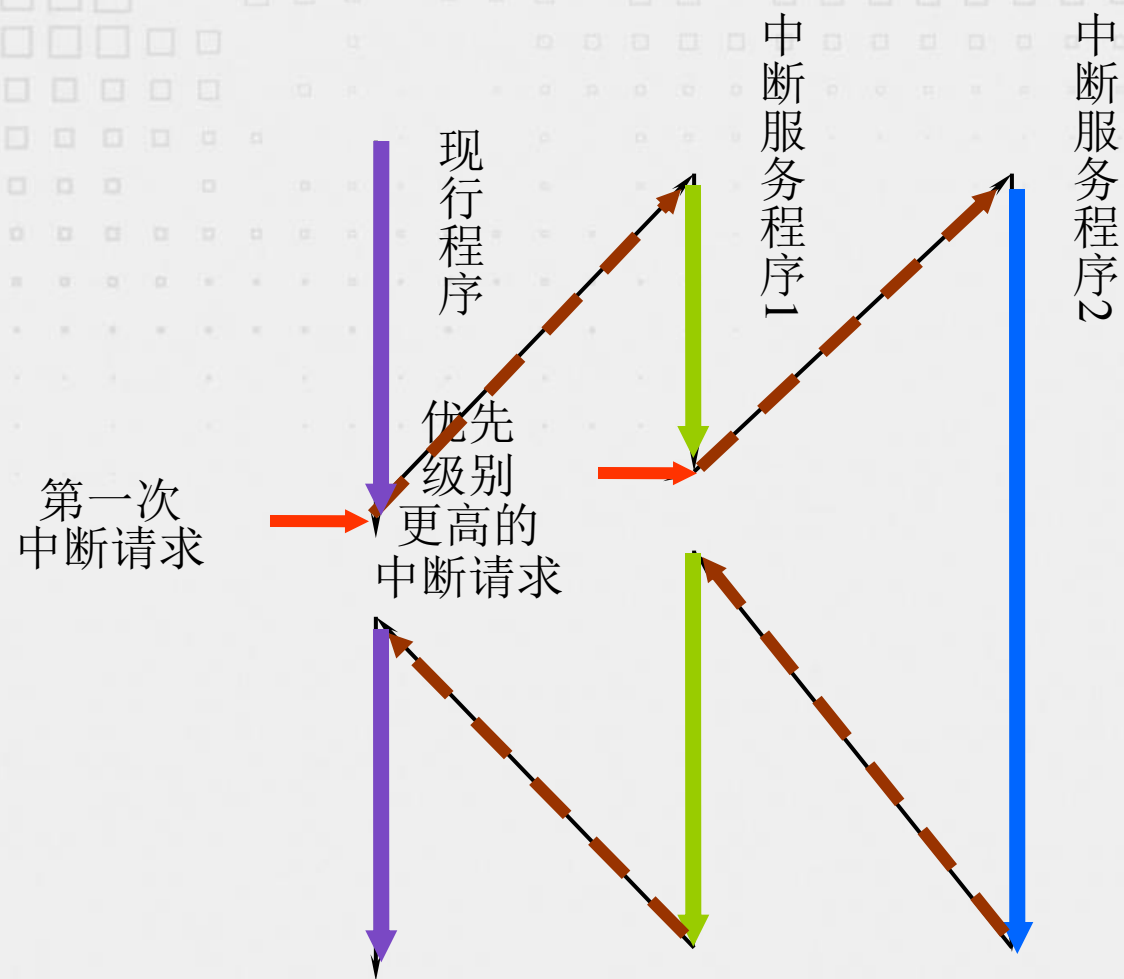
- 中断嵌套的层次可以有多层，越在里层的中断越急迫，优先级越高，因此优先得到CPU的服务。
- 要使计算机具有多重中断的能力，首先要能保护多个断点，先发生的中断请求的断点，先保护后恢复；堆栈的先进后出特点正好满足多重中断这一先后次序的需要。在CPU进入某一中断服务程序之后，系统必须处于开中断状态，否则中断嵌套是不可能实现的。





9.3.4

多重中断与中断屏蔽





- 2. 允许和禁止中断
 - 允许中断还是禁止中断是用CPU中的中断允许触发器控制的，当中断允许触发器被置“1”，则允许中断，当中断允许触发器被置“0”，则禁止中断。





- 2. 允许和禁止中断

- 允许中断即开中断，下列情况时应开中断：

- ✘ (1) 在中断服务程序执行完毕，恢复中断现场之后；
 - ✘ (2) 在多重中断的情况下，保护中断现场之后。

- 禁止中断即关中断，下列情况时应关中断：

- ✘ (1) 当响应某一级中断请求，不再允许被其他中断请求打断时；
 - ✘ (2) 在中断服务程序的保护和恢复现场之前。



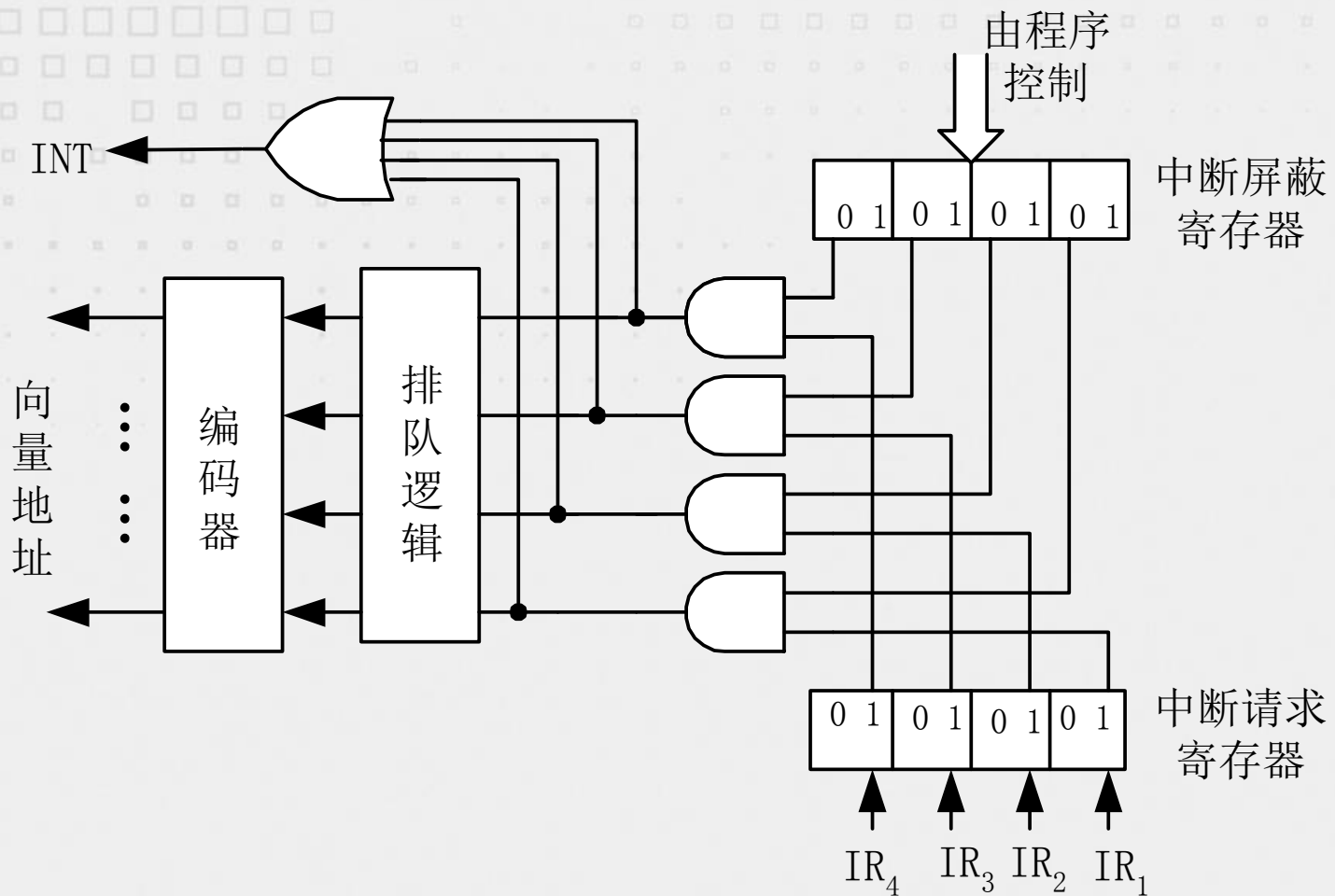


- 3.中断屏蔽
 - 中断源发出中断请求之后，这个中断请求并不一定能真正送到CPU去，在有些情况下，可以用程序方式有选择地封锁部分中断，这就是中断屏蔽。



9.3.4

多重中断与中断屏蔽





9.3.4

多重中断与中断屏蔽

• 3.中断屏蔽

- 如一个中断系统有16个中断源，每一个中断源按其优先级别赋予一个屏蔽字。
“0”表示开放，“1”表示屏蔽。

中断源的优先级	屏蔽字（16位）
1	111...111
2	011...111
3	001...111
⋮	⋮
15	000...011
16	000...001

- 第1级中断源的优先级别最高，它禁止本级和更低级的中断请求；第16级中断源的优先级别最低，它仅禁止本级的中断请求，而对其他高级的中断请求全部开放。





- 4.中断升级

- 中断屏蔽字的另一个作用是可以改变中断优先级，将原级别较低的中断源变成较高的级别，这实际上是一种动态改变优先级的方法。
- 这里所说的改变优先次序是指改变中断的处理次序。但是，中断处理次序是可以由屏蔽码来改变的，故把屏蔽码看成软排队器。中断处理次序可以不同于中断响应次序。





9.3.4

多重中断与中断屏蔽

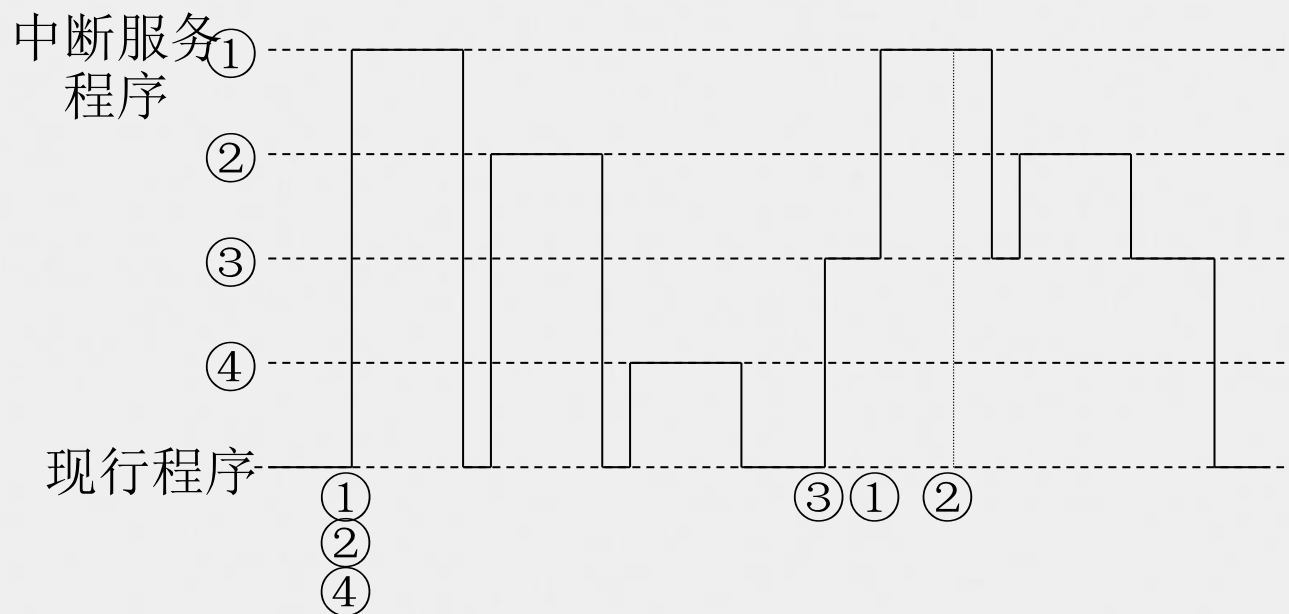
- 例如，某计算机的中断系统有4个中断源，每个中断源对应一个屏蔽码。中断响应的优先次序为1→2→3→4。中断的处理次序和中断的响应次序是一致的。

程序级别	屏蔽码			
	1级	2级	3级	4级
第1级	1	1	1	1
第2级	0	1	1	1
第3级	0	0	1	1
第4级	0	0	0	1





- 根据这一次序，可以看到CPU运动的轨迹，当多个中断请求同时出现时，处理次序与响应次序一致；当中断请求先后出现时，允许优先级别高的中断请求打断优先级别低的中断服务程序，实现中断嵌套。





9.3.4

多重中断与中断屏蔽

- 在不改变中断响应次序的条件下，通过改写屏蔽码可以改变中断处理次序，例如，要使中断处理次序改为1→4→3→2。

程序级别	屏蔽码			
	1级	2级	3级	4级
第1级	1	1	1	1
第2级	0	1	0	0
第3级	0	1	1	0
第4级	0	1	1	1



9.3.4 多重中断与中断屏蔽



-



- 中断全过程是指从中断源发出中断请求开始，CPU响应这个请求，现行程序被中断，转至中断服务程序，直至中断服务程序执行完毕，CPU 再返回原来的程序继续执行的整个过程。
- 中断全过程分为五个阶段：
 - 中断请求、中断判优、中断响应、中断处理、中断返回。





目录

1

9.1主机与外设的连接

2

9.2程序查询方式及其接口

3

9.3中断系统和程序中中断方式

4

9.4DMA方式及其接口

5

9.5通道控制方式



- 1. DMA方式的特点
 - 无论程序查询还是程序中断方式，主要的工作都是由CPU执行程序完成的，这需要花费时间，因此不能实现高速外设与主机的信息交换。
 - 直接存储器访问DMA方式是在**外设和主存储器之间**开辟一条“**直接数据通道**”，在不需CPU 干预也不需要软件介入的情况下在两者之间进行的高速数据传送方式。





- 在DMA传送方式中，对数据传送过程进行控制的硬件称为DMA控制器。当外设需要进行数据传送时，通过DMA控制器向CPU提出DMA传送请求，CPU响应之后将让出系统总线，由DMA控制器接管总线进行数据传送。





- DMA方式具有下列特点：
 - ① 它使主存与CPU的固定联系脱钩，主存既可被CPU访问，又可被外设访问。
 - ② 在数据块传送时，主存地址的确定，传送数据的计数等等都用硬件电路直接实现。
 - ③ 主存中要开辟专用缓冲区，及时供给和接收外设的数据。
 - ④ DMA传送速度快，CPU和外设并行工作，提高了系统的效率。
 - ⑤ DMA在开始前和结束后要通过程序和中断方式进行预处理和后处理。

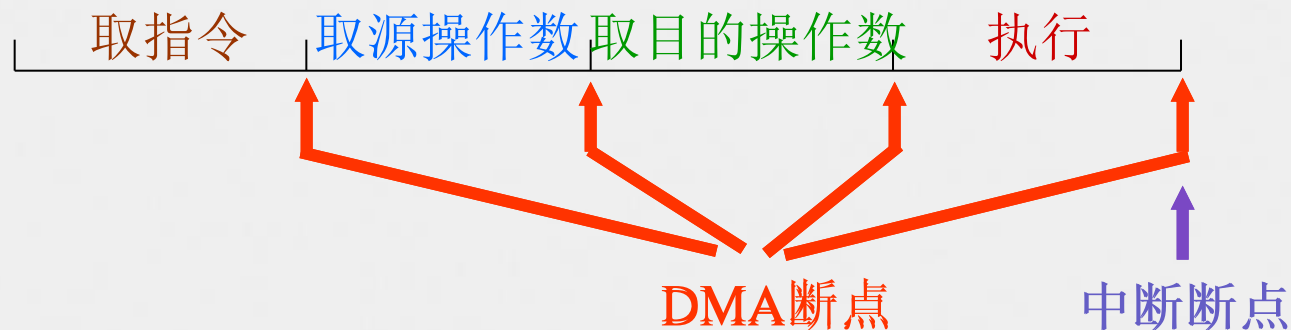




• 2. DMA和中断的区别

– 两者的重要区别为：

- ① 中断方式是程序切换，需要保护和恢复现场；而DMA方式除了开始和结尾时，不占用CPU的任何资源。
- ② 对中断请求的响应只能发生在每条指令执行完毕时；而对DMA请求的响应可以发生在每个机器周期结束时。





- 2. DMA和中断的区别

- 两者的重要区别为：

- ✘ ③ 中断传送过程需要CPU的干预；而DMA传送过程不需要CPU的干预，故数据传送速率非常高，适合于高速外设的成组数据传送。
 - ✘ ④ DMA请求的优先级高于中断请求。
 - ✘ ⑤ 中断方式具有对异常事件的处理能力；而DMA方式仅局限于完成传送信息块的I/O操作。





- 1.DMA控制器的功能
 - 在DMA传送过程中，DMA控制器将接管CPU的地址总线、数据总线和控制总线，CPU的主存控制信号被禁止使用。而当DMA传送结束后，将恢复CPU的一切权利并开始执行其操作。由此可见，DMA控制器必须具有控制系统总线的能力，即能够像CPU一样输出地址信号，接收或发出控制信号，输入或输出数据信号。





- DMA控制器在外设与主存之间直接传送数据期间，完全代替CPU进行工作，它的主要功能有：
 - (1) 接受外设发出的DMA请求，并向CPU发出总线请求；
 - (2) 当CPU响应此总线请求，发出总线响应信号后，接管对总线的控制，进入DMA操作周期；
 - (3) 确定传送数据的主存单元地址及传送长度，并能自动修改主存地址计数值和传送长度计数值；





- DMA控制器在外设与主存之间直接传送数据期间，完全代替CPU进行工作，它的主要功能有：
 - (4) 规定数据在主存与外设之间的传送方向，发出读/写或其他控制信号，并执行数据传送的操作。
 - (5) 向CPU报告DMA操作的结束。





- 2. DMA控制器的基本组成
 - (1) 主存地址计数器
 - (2) 传送长度计数器
 - (3) 数据缓冲寄存器
 - (4) DMA请求触发器
 - (5) 控制/状态逻辑
 - (6) 中断机构





- 3.DMA控制器的引出线
 - (1)地址总线
 - (2)数据总线
 - (3)控制数据传送方式的信号线
 - (4)DMA控制器与外设之间的联络信号线
 - (5)DMA控制器与CPU之间的联络信号线





- 4.DMA控制器的连接和传送
 - (1) 首先由外设向DMA控制器发出请求信号DREQ。
 - (2) DMA控制器向CPU发出总线请求信号HRQ。
 - (3) CPU向DMA控制器发出总线响应信号HLDA，此时，DMA控制器获取了总线的控制权。
 - (4) DMA控制器向外设发出DMA响应信号DACK，表示DMA控制器已控制了总线，允许外设与主存交换数据





- 4.DMA控制器的连接和传送
 - (5) DMA控制器按主存地址计数器的内容发出地址信号作为主存地址的选择，同时主存地址计数器的内容加1（或减1）。
 - (6) DMA控制器发出IOR信号到外设，将外设数据读入总线，同时发出MEMW信号，将数据总线的数据写入地址总线选中的主存单元。
 - (7) 传送长度计数器减1。



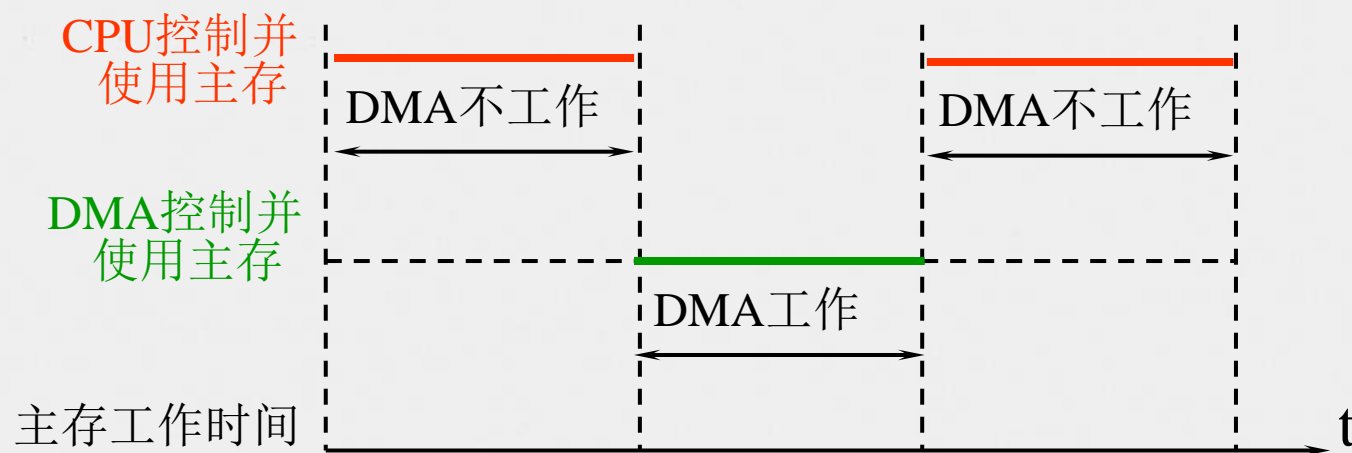


- 4.DMA控制器的连接和传送
 - 重复(5)(6)(7)步骤，直到字节计数器减到“0”为止，数据块的DMA方式传送工作宣告完成。这时，DMA控制器的HRQ降为低电平，总线控制权交还CPU。



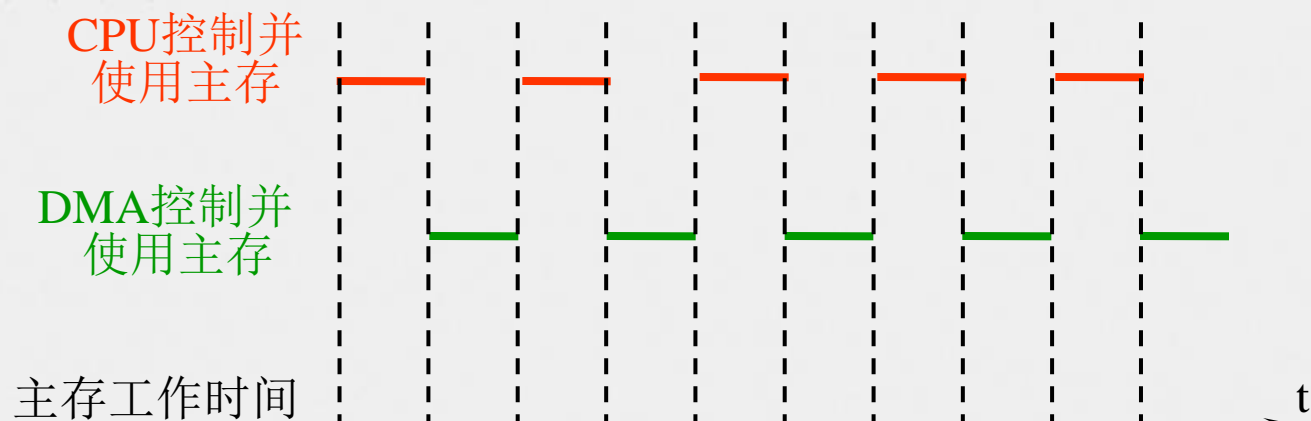


- 1. DMA传送方法
 - (1) CPU停止访问主存法



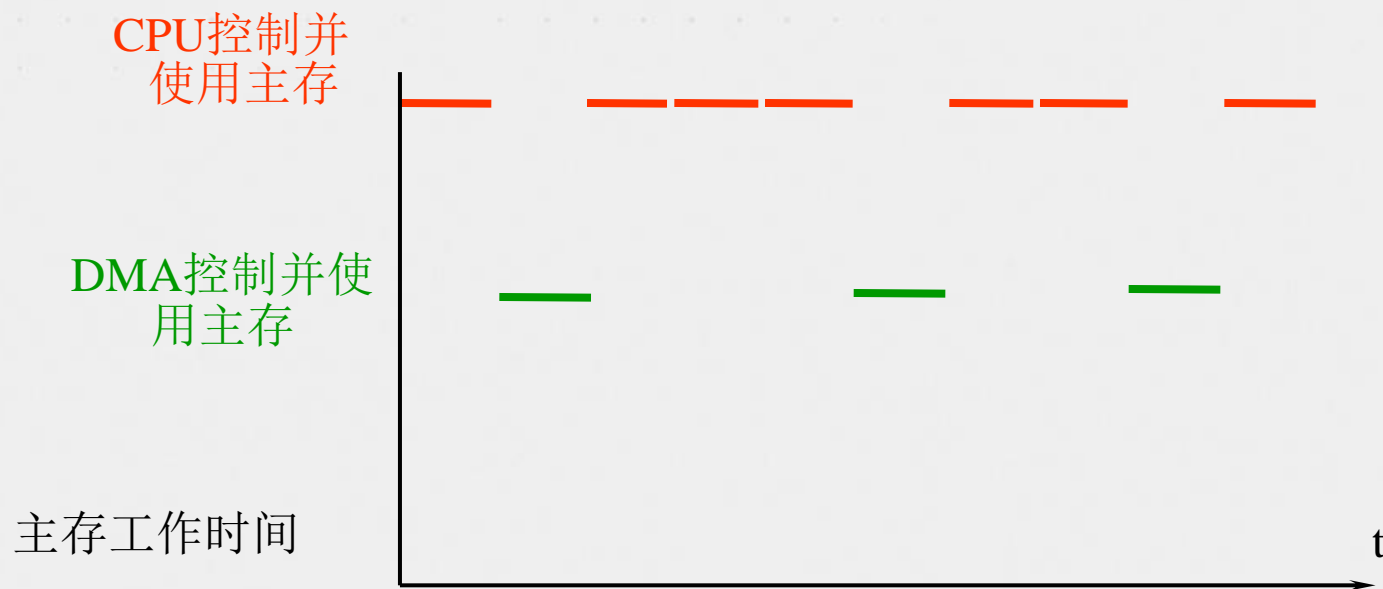


- 1. DMA传送方法
 - (2) 存储器分时法





- 1. DMA传送方法
 - (3) 周期挪用法





- 2. DMA传送过程

- (1) DMA预处理

- α 这是在DMA传送之前做的一些必要的准备工作，是由CPU来完成的。CPU 首先执行几条I/O指令，用于测试外设的状态、向DMA控制器的有关寄存器置初值、设置传送方向、启动该外部设备等。
 - α 在这些工作完成之后，CPU继续执行原来的程序，在外设准备好发送的数据（输入时）或接收的数据已处理完毕（输出时），外设向DMA控制器发DMA请求，再由DMA控制器向CPU发总线请求。





- 2. DMA传送过程

- (2) 数据传送

- ✘ DMA的数据传送可以是以单字节（或字）为基本单位，也可以以数据块为基本单位。对于以数据块为单位的传送，DMA 占用总线后的数据输入和输出操作都是通过循环来实现的。
 - ✘ 需要特别指出的是，这一循环不是由CPU执行程序实现的，而是由DMA控制器实现的。





- 2. DMA传送过程

- (3) DMA后处理

- α 当长度计数器计为0时，DMA操作结束，DMA控制器向CPU发中断请求，CPU 停止原来程序的执行，转去执行中断服务程序做DMA结束处理工作。





目录

1

9.1主机与外设的连接

2

9.2程序查询方式及其接口

3

9.3中断系统和程序中中断方式

4

9.4DMA方式及其接口

5

9.5通道控制方式



- 在大型计算机系统中，所连接的I/O设备数量多，输入/输出频繁，要求整体的速度快，单纯依靠主CPU采取中断和DMA等控制方式已不能满足要求。





- 1. 通道控制方式与DMA方式的区别
 - 通道控制方式是DMA方式的进一步发展，实质上，通道也是实现外设和主存之间直接交换数据的控制器。两者的主要区别在于：
 - ✘ ① DMA控制器是通过专门设计的硬件控制逻辑来实现对数据传送的控制；而通道则是一个具有特殊功能的处理器，它具有自己的指令和程序，通过执行一个通道程序实现对数据传送的控制，故通道具有更强的独立处理数据输入/输出的功能。
 - ✘ ② DMA控制器通常只能控制一台或少数几台同类设备；而一个通道则可以同时控制许多台同类或不同类的设备。





- 2. 通道的功能

- 通道在一定的硬件基础上利用软件手段实现对I/O的控制和传送，更多地免去了CPU的介入，从而使主机和外设的并行工作程度更高。当然，通道并不能完全脱离CPU，它还要受到CPU的管理，而且通道还应该向CPU报告自己的状态，以便CPU决定下一步的处理。





- 2. 通道的功能
 - ① 接受CPU的I/O指令，按指令要求与指定的外设进行联系。
 - ② 从主存取出属于该通道程序的通道指令，经译码后向设备控制器和设备发送各种命令。
 - ③ 实施主存和外设间的数据传送。
 - ④ 从外设获得设备的状态信息，形成并保存通道本身的状态信息，根据要求将这些状态信息送到主存的指定单元，供CPU使用。
 - ⑤ 将外设的中断请求和通道本身的中断请求按次序及时报告CPU。



THANK YOU

